Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	976	715/513.ccls.	USPAT	OR	OFF	2005/04/15 14:34
L2	184	715/515.ccls.	USPAT	OR	OFF	2005/04/15 14:34
L3	980	707/203.ccls.	USPAT	OR	OFF	2005/04/15 14:36
L4	735	707/202.ccls.	USPAT	OR	OFF	2005/04/15 14:39
L5	319	711/161.ccls.	USPAT	OR	OFF	2005/04/15 14:44
S142	542	711/162.ccls.	USPAT	OR	OFF	2005/04/15 11:49
S143	671	707/202.ccls.	USPAT	OR	OFF	2005/04/15 11:49
S144	880	707/203.ccls.	USPAT	OR	OFF	2005/04/15 11:49
S145	3778741	primary final name file recover$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/05/24 16:08
S146	2	primary final name file recover$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	SAME	OFF	2004/05/24 16:08
S147	1	recover$3 file temporary name final	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:28
S148	520	recover$3 file list	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:54
S149	3	S148 automatic backup	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:36
S150	2	S148 sav$3 copy document	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:39
S151	2	S148 sav$3 crash document	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:39
S152	3	S148 sav$3 crash	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 11:40
S155	663	711/162.ccls.	USPAT	OR	OFF	2005/04/15 11:49

S156	806	707/204.ccls.	USPAT	OR	OFF	2005/04/15 11:49
S157	202	713/165.ccls.	USPAT	OR	OFF	2005/04/15 11:50
S158	507	715/530.ccls.	USPAT	OR	OFF	2005/04/15 11:50
S160	18	S148 failure computer	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:05
S161	1	S148 temporary final name	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:05
S162	1	S148 temporary final name recover$3	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:23
S163	0	crash recover$3 (html xml sgml)	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:24
S164	1	crash recover$3 (html or xml or sgml)	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:24
S165	3	crash recover$3 (web adj page)	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:41
S166	0	crash recover$3 final name temporary	USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2005/04/15 12:41

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+recover??? <paragraph> +file <paragraph> +name <paragr SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
recover??? paragraph file paragraph name paragraph temporary paragraph final

Found **232** of **153,034**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

21 Ada development system technical and performance requirements (with rationale)

Donald G. Krantz

December 1990 **Proceedings of the conference on TRI-ADA '90**

Full text available: pdf(1.85 MB) Additional Information: full citation, abstract, references

This paper discusses requirements for Ada1 compilers and associated tools used for real-time embedded weapons systems (EWS) development. The requirements have been developed over a period of several years by embedded systems developers at Honeywell Inc. and Alliant Techsystems Inc. Requirements for the run time system, compiler-generated code, and host tools such as linkers are presented. A short rationale statement is provided with each specific requirement.

22 Risks to the public in computers and related systems

Peter G. Neumann

January 1990 **ACM SIGSOFT Software Engineering Notes**, Volume 15 Issue 1

Full text available: pdf(2.11 MB) Additional Information: full citation

23 Human factors challenges in creating a principal support office system—the speech filing system approach

John D. Gould, Stephen J. Boies

October 1983 **ACM Transactions on Information Systems (TOIS)**, Volume 1 Issue 4

Full text available: pdf(1.65 MB) Additional Information: full citation, references, citings, index terms

24 Technical reports

SIGACT News Staff

January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Full text available: pdf(5.28 MB) Additional Information: full citation

25 A single-pass syntax-directed front end for Ada

T. P. Baker

June 1982 **ACM SIGPLAN Notices , Proceedings of the 1982 SIGPLAN symposium on Compiler construction**, Volume 17 Issue 6

Additional Information: full citation, abstract, references, citings, index

Full text available: pdf(842.95 KB) terms

This paper describes the front-end processor of an Ada compiler that is under development at Florida State University. The compiler is coded in Pascal, to execute on a CDC Cyber system, and is presently targeted to the Z8000 microprocessor architecture. Owing at least in part to the peculiar origins and changing goals of this project, the front end processor is rather unlike those of the other Ada compilers of which we know. Perhaps its most distinctive feature is that it operates in one pa ...

26 The evolution of the Sperry Univac 1100 series: a history, analysis, and projection

B. R. Borgerson, M. L. Hanson, P. A. Hartley

January 1978 **Communications of the ACM**, Volume 21 Issue 1

Full text available: pdf(1.89 MB) Additional Information: full citation, abstract, citings, index terms

The 1100 series systems are Sperry Univac's large-scale mainframe computer systems. Beginning with the 1107 in 1962, the 1100 series has progressed through a succession of eight compatible computer models to the latest system, the 1100/80, introduced in 1977. The 1100 series hardware architecture Is based on a 36-bit word, ones complement structure which obtains one operand from storage and one from a high-speed register, or two operands from high-speed registers. The 1100 Operating System ...

Keywords: 1100 computer series, computer architecture, data management systems, end user facilities, executive control software, multiprocessing, multiprogramming, operating system, programming languages

27 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

28 Risks to the public in computer systems

Peter G. Neumann

October 1986 **ACM SIGSOFT Software Engineering Notes**, Volume 11 Issue 5

Full text available: pdf(2.19 MB) Additional Information: full citation, index terms

29 Cooperative transaction hierarchies: transaction support for design applications

Marian H. Nodine, Stanley B. Zdonik

July 1992 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 1 Issue 1

Full text available: pdf(2.20 MB) Additional Information: full citation, abstract, references, citings

Traditional atomic and nested transactions are not always well-suited to cooperative applications, such as design applications. Cooperative applications place requirements on the database that may conflict with the serializability requirement. They require transactions to be long, possibly nested, and able to interact with each other in a structured

way. We define a transaction framework, called a *cooperative transaction hierarchy,* that allows us to relax the requirement for atomic, seria ...

Keywords: cooperation, deadlock detection, design transactions, non-serializability, transaction hierarchies, transaction synchronization, version management

30 The mobile programming system: STAGE2

W. M. Waite

July 1970 **Communications of the ACM**, Volume 13 Issue 7

Full text available: pdf(698.40 KB) Additional Information: full citation, abstract, references, citings

STAGE2 is the second level of a bootstrap sequence which is easily implemented on any computer. It is a flexible, powerful macro processor designed specifically as a tool for constructing machine-independent software. In this paper the features provided by STAGE2 are summarized, and the implementation techniques which have made it possible to have STAGE2 running on a new machine with less than one man-week of effort are discussed. The approach has been successful on over 15 machines of wide ...

Keywords: bootstrapping, implementation techniques, machine independence, macro processing, programming languages

31 Status report of the graphic standards planning committee

Computer Graphics staff

August 1979 **ACM SIGGRAPH Computer Graphics**, Volume 13 Issue 3

Full text available: pdf(15.01 MB) Additional Information: full citation, references, citings

32 Distributed file systems: concepts and examples

Eliezer Levy, Abraham Silberschatz

December 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 4

Full text available: pdf(5.33 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is implemented as part of the operating system of each of the connected computers. This paper establishes a viewpoint that emphasizes the dispersed structure and decentralization of both data and con ...

33 UIDs as internal names in a distributed file system

Paul J. Leach, Bernard L. Stumpf, James A. Hamilton, Paul H. Levine

August 1982 **Proceedings of the first ACM SIGACT-SIGOPS symposium on Principles of distributed computing**

Full text available: pdf(722.48 KB) Additional Information: full citation, abstract, references, citings, index terms

The use of UIDs as internal names in an operating system for a local network is discussed. The use of internal names in other distributed systems is briefly surveyed. For this system, UIDs were chosen because of their intrinsic location independence and because they seemed to lend themselves to a clean structure for the operating system nucleus. The problems created by UIDs were: generating UIDs; locating objects; supporting multiple versions of objects; replicating objects; and losing obje ...

34 Deciding when to forget in the Elephant file system

Douglas S. Santry, Michael J. Feeley, Norman C. Hutchinson, Alistair C. Veitch, Ross W.

Carton, Jacob Ofir
December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles**, Volume 33 Issue 5

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings, index terms

Modern file systems associate the deletion of a file with the immediate release of storage, and file writes with the irrevocable change of file contents. We argue that this behavior is a relic of the past, when disk storage was a scarce resource. Today, large cheap disks make it possible for the file system to protect valuable data from accidental delete or overwrite.This paper describes the design, implementation, and performance of the Elephant file system, which automatically retains all impo ...

35 Decentralized storage systems: Taming aggressive replication in the Pangaea wide-area file system

Yasushi Saito, Christos Karamanolis, Magnus Karlsson, Mallik Mahalingam
December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Full text available: pdf(1.93 MB) Additional Information: full citation, abstract, references

Pangaea is a wide-area file system that supports data sharing among a community of widely distributed users. It is built on a symmetrically decentralized infrastructure that consists of commodity computers provided by the end users. Computers act autonomously to serve data to their local users. When possible, they exchange data with nearby peers to improve the system's overall performance, availability, and network economy. This approach is realized by aggressively creating a replica of a file w ...

36 Special issue: Game-playing programs: theory and practice

M. A. Bramer
April 1972 **ACM SIGART Bulletin**, Issue 80

Full text available: pdf(9.23 MB) Additional Information: full citation, abstract

This collection of articles has been brought together to provide SIGART members with an overview of Artificial Intelligence approaches to constructing game-playing programs. Papers on both theory and practice are included.

37 The design and implementation of INGRES

Michael Stonebraker, Gerald Held, Eugene Wong, Peter Kreps
September 1976 **ACM Transactions on Database Systems (TODS)**, Volume 1 Issue 3

Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, references, citings, index terms

The currently operational (March 1976) version of the INGRES database management system is described. This multiuser system gives a relational view of data, supports two high level nonprocedural data sublanguages, and runs as a collection of user processes on top of the UNIX operating system for Digital Equipment Corporation PDP 11/40, 11/45, and 11/70 computers. Emphasis is on the design decisions and tradeoffs related to (1) structuring the system into processes, (2) embedding one command ...

Keywords: concurrency, data integrity, data organization, data sublanguage, database optimization, nonprocedural language, protection, query decompositon, query language, relational database

38 Introduction to a system for distributed databases (SDD-1)

J. B. Rothnie, P. A. Bernstein, S. Fox, N. Goodman, M. Hammer, T. A. Landers, C. Reeve, D. W. Shipman, E. Wong
March 1980 **ACM Transactions on Database Systems (TODS)**, Volume 5 Issue 1

Full text available: pdf(1.23 MB) Additional Information: full citation, abstract, references, citings, index

terms

The declining cost of computer hardware and the increasing data processing needs of geographically dispersed organizations have led to substantial interest in distributed data management. SDD-1 is a distributed database management system currently being developed by Computer Corporation of America. Users interact with SDD-1 precisely as if it were a nondistributed database system because SDD-1 handles all issues arising from the distribution of data. These issues include distributed concurr ...

Keywords: concurrency control, database reliability, distributed database system, query processing, relational data model

39 Algorithm 719; Multiprecision translation and execution of FORTRAN programs

David H. Bailey

September 1993 **ACM Transactions on Mathematical Software (TOMS)**, Volume 19 Issue 3

Full text available: pdf(2.03 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes two Fortran utilities for multiprecision computation. The first is a package of Fortran subroutines that perform a variety of arithmetic operations and transcendental functions on floating point numbers of arbitrarily high precision. This package is in some cases over 200 times faster than that of certain other packages that have been developed for this purpose. The second utility is a translator program, which facilitates the conversion of ordinary Fortran p ...

Keywords: multiple-precision computation, multiprecision arithmetic

40 The design and implementation of a log-structured file system

Mendel Rosenblum, John K. Ousterhout

February 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 1

Full text available: pdf(1.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper presents a new technique for disk storage management called a log-structured file system. A log-structured file system writes all modifications to disk sequentially in a log-like structure, thereby speeding up both file writing and crash recovery. The log is the only structure on disk; it contains indexing information so that files can be read back from the log efficiently. In order to maintain large free areas on disk for fast writing, we divide the log into

Keywords: Unix, disk storage management, fast crash recovery, file system organization, file system performance, high write performance, log-structured, logging

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

Results for "((crash <paragraph> recover??? <paragraph> xml)<in>metadata)" e-mail
Your search matched 0 of 1144303 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

» View Session History

» New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Modify Search

((crash <paragraph> recover??? <paragraph> xml)<in>metadata)

Check to search only within this results set

Display Format: Citation Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revisir

Indexed by Inspec

© Copyright 2005 IEEE –

Results for "((crash <paragraph> recover??? <paragraph> final <paragraph> name <paragraph> temporary) <in>metadata)" e-mail

Your search matched **0** of **1144315** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» View Session History

» New Search

Modify Search

((crash <paragraph> recover??? <paragraph> final <paragraph> name <paragraph> >>

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revisir

Indexed by Inspec

© Copyright 2005 IEEE –

	1	Document ID	Issue Date	Title	Current OR	Inventor
1		US 4566066 A	19860121	Securities valuation system	705/36	Towers; Frederic C.
2		US 4334270 A	19820608	Securities valuation system	705/36	Towers; Frederic C.
3		NB8503603 1	19850301	Automatic Recovery From Media Damage While Loading Objects		

	1	Document ID	Issue Date	Title	Current OR	Inventor
1		NN9105110	19910501	SNA/DS Data Processing End Node Implementation.		
2		NN83112908	19831101	A Journal Architecture for a Segmented, Single Level Store		
3	X	US 6826729 B1	20041130	Gallery user interface controls	715/837	Giesen; Ronald Stephen et al.

	1	Document ID	Issue Date	Title	Current OR	Inventor
1		US 6836855 B2	20041228	Recovery from data fetch errors in hypervisor code	714/9	Arndt; Richard Louis
2		US 6785674 B2	20040831	System and method for structuring data in a computer system	707/3	Vu; Jonathan
3		US 6638314 B1	20031028	Method of web crawling utilizing crawl numbers	715/513	Meyerzon; Dmitriy et al.
4		US 6446090 B1	20020903	Tracker sensing method for regulating synchronization of audit files between primary and secondary hosts	707/201	Hart; Donald Ralph
5		US 6430698 B1	20020806	Virtual distributed home agent protocol	714/4	Khalil; Mohamed et al.
6		US 6430577 B1	20020806	System and method for asynchronously receiving multiple packets of audit data from a source databased host in a resynchronization mode and asynchronously writing the data to a target host	707/201	Hart; Donald Ralph

	1	Document ID	Issue Date	Title	Current OR	Inventor
7		US 6192401 B1	20010220	System and method for determining cluster membership in a heterogeneous distributed system	709/220	Modiri; Ramin et al.
8		US 6173415 B1	20010109	System for scalable distributed data structure having scalable availability	714/7	Litwin; Witold et al.
9		US 6122754 A	20000919	Method and system for data recovery using a distributed and scalable data structure	714/4	Litwin; Witold et al.
10		US 5860090 A	19990112	Append-only storage in a disk array using striping and parity caching	711/113	Clark; Thomas K.
11		US 5590318 A	19961231	Method and system for tracking files pending processing	707/202	Zbikowski; Mark et al.
12		WO 3075141 A1	20030912	COMPUTER FUNCTION STATUS STORAGE METHOD AT A STOPPAGE OF ELECTRIC CURRENT		PARK, JAE SEUNG et al.
13		NN9010445	19901001	Combined File Operations for Simplified Recovery.		
14		NA8908448	19890801	Fast Two-Pass Write		

	1	Document ID	Issue Date	Title	Current OR	Inventor
15		NN8806158	19880601	Differential Index Management With Low Priority Batch Updating in Database Management Systems		
16		NN85057236	19850501	Diagnostic Error Logging in a Copier Monitoring System		
17		NN83112908	19831101	A Journal Architecture for a Segmented, Single Level Store		
18		NN8006356	19800601	Recording and Notification of Errors in a Document Transmission System. June 1980.		

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+recover??? <paragraph> +file <paragraph> +name <paragr SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
recover??? paragraph file paragraph name paragraph temporary paragraph final

Found **232** of **153,034**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 File servers for network-based distributed systems

Liba Svobodova

December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4

Full text available: pdf(4.23 MB) Additional Information: full citation, references, citings, index terms, review

2 Pen computing: a technology overview and a vision

André Meyer

July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

Full text available: pdf(5.14 MB) Additional Information: full citation, abstract, citings, index terms

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.Starting with a short historic ...

3 Proceedings of the SIGNUM conference on the programming environment for development of numerical software

March 1979 **ACM SIGNUM Newsletter**, Volume 14 Issue 1

Full text available: pdf(5.02 MB) Additional Information: full citation

4 A coherent distributed file cache with directory write-behind

Timothy Mann, Andrew Birrell, Andy Hisgen, Charles Jerian, Garret Swart

May 1994 **ACM Transactions on Computer Systems (TOCS)**, Volume 12 Issue 2

Full text available: pdf(3.21 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Extensive caching is a key feature of the Echo distributed file system. Echo client machines maintain coherent caches of file and directory data and properties, with write-behind (delayed write-back) of all cached information. Echo specifies ordering constraints on this write-behind, enabling applications to store and maintain consistent data structures in the file system even when crashes or network faults prevent some writes from being completed. In this paper we describe ...

Keywords: coherence, file caching, write-behind

5 Disconnected operation in the Coda File System
James J. Kistler, M. Satyanarayanan
February 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 1

Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Disconnected operation is a mode of operation that enables a client to continue accessing critical data during temporary failures of a shared data repository. An important, though not exclusive, application of disconnected operation is in supporting portable computers. In this paper, we show that disconnected operation is feasible, efficient and usable by describing its design and implementation in the Coda File System. The central idea behind our work is that cachi ...

Keywords: disconnected operation, hoarding, optimistic replication, reintegration, second-class replication, server emulation

6 Principles of transaction-oriented database recovery
Theo Haerder, Andreas Reuter
December 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 4

Full text available: pdf(2.48 MB) Additional Information: full citation, references, citings, index terms, review

7 Automatic high-quality reengineering of database programs by abstraction, transformation and reimplementation
Yossi Cohen, Yishai A. Feldman
July 2003 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 12 Issue 3

Full text available: pdf(245.97 KB) Additional Information: full citation, abstract, references, index terms

Old-generation database models, such as the indexed-sequential, hierarchical, or network models, provide record-level access to their data, with all application logic residing in the hosting program. In contrast, relational databases can perform complex operations, such as filter, aggregation, and join, on multiple records without an external specification of the record-access logic. Programs written for relational databases attempt to move as much of the application logic as possible into the d ...

Keywords: Database program reengineering, query graphs, temporal abstraction, the plan calculus

8 UIO: a uniform I/O system interface for distributed systems
David R. Cheriton
January 1987 **ACM Transactions on Computer Systems (TOCS)**, Volume 5 Issue 1

Full text available: pdf(3.20 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A uniform I/O interface allows programs to be written relatively independently of specific I/O services and yet work with a wide variety of the I/O services available in a distributed environment. Ideally, the interface provides this uniform access without excessive complexity in the interface or loss of performance. However, a uniform interface does not arise from careful design of individual system interfaces alone; it requires explicit definition. In this paper, the UIO (unifo ...

9 Co-array Fortran for parallel programming
Robert W. Numrich, John Reid

- August 1998 **ACM SIGPLAN Fortran Forum**, Volume 17 Issue 2

Full text available: pdf(1.94 MB) Additional Information: full citation, abstract, citings, index terms

Co-Array Fortran, formerly known as F--, is a small extension of Fortran 95 for parallel processing. A Co-Array Fortran program is interpreted as if it were replicated a number of times and all copies were executed asynchronously. Each copy has its own set of data objects and is termed an image. The array syntax of Fortran 95 is extended with additional trailing subscripts in square brackets to give a clear and straightforward representation of any access to data that is spread across ...

10 A cryptographic file system for UNIX

Matt Blaze

December 1993 **Proceedings of the 1st ACM conference on Computer and communications security**

Full text available: pdf(955.62 KB) Additional Information: full citation, abstract, references, citings, index terms

Although cryptographic techniques are playing an increasingly important role in modern computing system security, user-level tools for encrypting file data are cumbersome and suffer from a number of inherent vulnerabilities. The Cryptographic File System (CFS) pushes encryption services into the file system itself. CFS supports secure storage at the system level through a standard Unix file system interface to encrypted files. Users associate a cryptographic key with the directories ...

11 Assembly instruction level reverse execution for debugging

Tankut Akgul, Vincent J. Mooney III

April 2004 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 13 Issue 2

Full text available: pdf(1.18 MB) Additional Information: full citation, abstract, references, index terms

Assembly instruction level reverse execution provides a programmer with the ability to return a program to a previous state in its execution history via execution of a "reverse program." The ability to execute a program in reverse is advantageous for shortening software development time. Conventional techniques for recovering a state rely on saving the state into a record before the state is destroyed. However, state-saving causes significant memory and time overheads during forward execution.Th ...

Keywords: Debugging, reverse code generation, reverse execution

12 Cooperative visual manipulation of music notation

P. Bellini, P. Nesi, M. B. Spinu

September 2002 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 9 Issue 3

Full text available: pdf(3.42 MB) Additional Information: full citation, abstract, references, citings, index terms

As computer technologies and their potential emerging applications spread out, new needs have been detected for computer-based applications of music; cooperative music notation editing both in orchestras and music schools is one of them. This article is the only public document describing the details of cooperative work on music notation of MOODS (Music Object Oriented Distributed System). MOODS is a synchronous real-time cooperative editor for music scores. Its architecture includes mechanisms ...

Keywords: Collaboration of music notation editing, additional command list, collaborative systems, computer-supported cooperative work, consistency control, cooperative music, distributed music, electronic lectern, neutral version, selective undo, user interface management systems

13 A formal approach to undo operations in programming languages

George B. Leeman
January 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 1

Full text available: pdf(2.74 MB) Additional Information: full citation, abstract, references, citings, index terms

A framework is presented for adding a general Undo facility to programming languages. A discussion of relevant literature is provided to show that the idea of Undoing pervades several areas in computer science, and even other disciplines. A simple model of computation is introduced, and it is augmented with a minimal amount of additional structure needed for recovery and reversal. Two different interpretations of Undo are motivated with examples. Then, four primitives are defined in a langu ...

14 A user's viewpoint on the Programmer's Workbench

M. H. Bianchi, J. L. Wood
October 1976 **Proceedings of the 2nd international conference on Software engineering**

Full text available: pdf(841.32 KB) Additional Information: full citation, abstract, references, citings, index terms

The Programmer's Workbench boasts a broad set of highly useful features aimed at the application program developer. It claims to be a "human-end" computer providing tools and services to ease the load on the application system designer, programmer, documenter, tester, and delivery personnel. This paper shows the benefits of using the PWB tools, individually and in combination. Through specific examples drawn from the history of a software project, evidence is given that the use ...

Keywords: Programming aids, Software development, UNIX

15 Relational Database Systemsr

Won Kim
September 1979 **ACM Computing Surveys (CSUR)**, Volume 11 Issue 3

Full text available: pdf(2.67 MB) Additional Information: full citation, references, citings, index terms

16 Distributed operating systems

Andrew S. Tanenbaum, Robbert Van Renesse
December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available: pdf(5.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

17 Special issue: AI in engineering

D. Sriram, R. Joobbani
January 1985 **ACM SIGART Bulletin**, Issue 91

Full text available: pdf(8.79 MB) Additional Information: full citation, abstract

The papers in this special issue were compiled from responses to the announcement in the July 1984 issue of the SIGART newsletter and notices posted over the ARPAnet. The interest being shown in this area is reflected in the sixty papers received from over six countries. About half the papers were received over the computer network.

18 Columns: Risks to the public in computers and related systems

Peter G. Neumann
March 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 2
Full text available: pdf(1.54 MB) Additional Information: full citation

19 A Status Report on Computing Algorithms for Mathematical Programming
William W. White
September 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 3
Full text available: pdf(3.02 MB) Additional Information: full citation, references, citings, index terms

20 Risks to the public in computers and related systems
Peter G. Neumann
April 1990 **ACM SIGSOFT Software Engineering Notes**, Volume 15 Issue 2
Full text available: pdf(2.07 MB) Additional Information: full citation, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player